# Self-supervised learning on millions of pre-mRNA sequences improves sequence-based RNA splicing prediction

Ken Chen[1], Yue Zhou[2], Maolin Ding[1], Yu Wang[2], Zhixiang Ren[2,*], and Yuedong Yang[1,3,*]

[1] School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China,
[2] Peng Cheng Laboratory, Shenzhen, China,
[3] Key Laboratory of Machine Intelligence and Advanced Computing (Sun Yat-sen University), Ministry of Education, China.
* To whom correspondence should be addressed. Email: renzhx@pcl.ac.cn, yangyd25@mail.sysu.edu.cn

## ABSTRACT

RNA splicing is an important post-transcriptional process of gene expression in eukaryotic organisms. Here, we developed a novel language model, SpliceBERT, pre-trained on the precursor messenger RNA sequences of 72 vertebrates to improve sequence-based modelling of RNA splicing. SpliceBERT is capable of generating embeddings that preserve the evolutionary information of nucleotides and functional characteristics of splice sites. Moreover, the pre-trained model can be utilized to prioritize potential splice-disrupting variants in an unsupervised manner based on genetic variants' impact on the output of SpliceBERT for sequence context. Benchmarked on a multi-species splice site and a human branchpoint prediction task, SpliceBERT outperformed not only conventional baseline models but also other language models pretrained only on the human genome. Our study highlighted the importance of unsupervised learning with genomic sequences of multiple species and indicated that language models were promising approaches to decipher the determinants of RNA splicing.

## INTRODUCTION

Ribonucleic acids (RNA) splicing is a fundamental post-transcriptional process in eukaryotic gene expression, which removes introns from precursor messenger RNAs (pre-mRNAs) and ligates exons into mature mRNAs. Though the mechanism underlying RNA splicing is complex, a variety of studies have found that some key determinants of splicing are encoded in DNA sequences (1–3). Therefore, deciphering splicing codes from RNA sequences by computational models is a promising approach and will facilitate the interpretation of genetic variants affecting RNA splicing.

Early studies mainly aimed to identify short sequence motifs related to splicing, including exonic splicing enhancers (4), branchpoints (5) and other splicing factors (6) with statistical models. Benefiting from accumulated high-throughput sequencing data, other studies were able to employ more advanced machine learning and deep learning models to directly predict RNA splicing events from sequences. For example, HAL (7) is a model trained on alternative splicing events from millions of random sequences for predicting the change of exon skipping and 5' alternative splicing induced by genetic variations. SPANR (3) is a Bayesian network model for predicting the Percent Spliced In (PSI or Ψ) of alternatively spliced exons. More recently, deep learning models like MMSplice (8), SpliceAI (9) and Pangolin (10) employed deep convolutional neural networks (CNN) to predict alternative splicing events, splice sites or splice site usage. These methods achieved superior performance as compared to earlier ones and have been widely utilized to analyze aberrant RNA splicing events caused by genetic variants. Though significant progress has been made in this field, there is still room for further improvements. For instance, these methods were developed to predict splice sites or alternative splicing primary sequences using primary sequences, while evolutionary information was not considered. Besides, other important splicing regulators like branchpoints (BPs) were less studied. Unlike splice sites, which can be reliably detected from RNA-seq data (11, 12), BPs are much more difficult to be experimentally identified (13), making it challenging to train deep learning models for BP prediction due to the lack of adequate high-confidence BPs.

To alleviate the problem of insufficient data, self-supervised learning (SSL) strategy utilized in large language models (LLM) like BERT (14) can be adopted. The SSL strategy has been utilized to learn representations of protein (15, 16), non-coding RNA (17) and prokaryote genome (18) sequences. These models were pre-trained on millions of sequences from a large number of species and therefore captured the evolutionary information that is critical for modelling. Unfortunately, these methods cannot be directly applied to study RNA splicing because eukaryotic protein-coding RNA sequences are very different from non-coding RNAs or prokaryote genome sequences. Though there are other genome language models that can be adopted to study RNA splicing (19, 20), they were pre-trained on the human genome, and thus it remains unclear whether pre-training BERT-like models on sequences from many species could improve sequence-based RNA splicing prediction.

Here, we developed a BERT-based model named SpliceBERT for studying RNA splicing. SpliceBERT was pre-trained on more than 2 million precursor messenger RNA (pre-mRNA) sequences from 72 vertebrates with gene annotations. We found that compared with models pre-trained on the human genome, SpliceBERT can better preserve evolutionary information in nucleotide embeddings. The hidden states and attention weights in SpliceBERT could reflect the functional characteristics of splice sites and branchpoints. Besides, we observed that splice-disrupting variants can induce a larger impact on their context than those with little effect on splicing and thus SpliceBERT could serve as an unsupervised variant effect predictor for variants related to RNA splicing. On a multi-species splice site prediction task and a human branchpoint prediction task, SpliceBERT exhibited superior performance to conventional models without pre-training or other

language models pre-trained on only the human genome, further indicating the importance of SSL on the gene sequences of many species. The source codes and datasets of SpliceBERT are available at https://github.com/biomed-AI/SpliceBERT.

## MATERIAL AND METHODS

### The SpliceBERT model

***Pre-training data*** We extracted the pre-mRNA sequences from 72 vertebrate genomes for pre-training. The reference genomes in fasta format and gene annotations in gtf format were downloaded from UCSC genome browser (21). The versions of the genomes and annotations are available in Table S1. The pre-mRNA sequences were extracted from genome sequences using "bedtools getfasta" (22) based on gene annotations and overlapping transcripts were merged. In this way, we obtained a total number of 2011517 pre-mRNA sequences covering about 65 billion nucleotides (nt). We randomly reserved 50000 sequences for validation and pre-trained SpliceBERT using the other sequences.

***Model architecture*** We adopted the BERT architecture (14) to build SpliceBERT. SpliceBERT is consisted of 6 Transformer encoder layers, where the hidden layer size and the number of attention heads are 512 and 16, respectively. The position information of the nucleotides in each sequence are encoded using the one-hot positional embedding method. SpliceBERT has a total of about 19.4 million parameters.

***Tokenization*** Previous pre-trained models for DNA sequences mainly adopted the k-mer tokenization (19, 20, 23) or the Byte Pair Encoding (BPE) tokenization (24) strategy. However, in this study, we simply used each nucleotide type (A, G, C, T/U) as a single token for the ease of getting embeddings for individual nucleotides. The [CLS] token (classification) and [SEP] token (separator) were added to the beginning and the end of each sequence input to SpliceBERT, which is a routine operation in most BERT models. The [MASK] token was used to represent the masked tokens in MLM task.

***Pre-training by masked language modelling task*** We pre-trained SpliceBERT by the masked language modelling (MLM) task, which is a self-supervised learning strategy to pre-train language models. Specifically, for each input sequence, we randomly picked 15% of the nucleotides and 80%, 10%, 10% of which were replaced with the special token "[MASK]", random nucleotides, and original nucleotides (they were kept unchanged), respectively. During pre-training, the model was optimized to predict the correct base type of the original nucleotides. Intuitively, the model was required to learn the dependencies between the nucleotides in the input sequences to recover the corrupted tokens. In this way, we expected the model could capture the complex logic that may be encoded in the DNA/RNA sequences.

***Model implementation and pre-training*** We implemented SpliceBERT in PyTorch with the Huggingface transformer library (v4.24.0). The maximum length of input sequences was set to 1024. Since most pre-mRNA sequences are longer than 1024nt, fragments no longer than 1024nt were randomly sampled for pre-training. We used the cross-entropy loss as the objective to optimize the model. The AdamW optimizer was used and the initial learning rate was set to 0.0001. The learning rate will be reduced by a factor of 2 when model's performance on the validation data stops improving for 3 epochs. The model was trained in the automatic mixed precision (AMP) mode to decrease the consumption of memory and speed up model training. In practice, we pre-trained SpliceBERT in two stages. In the first stage, we used fixed-length sequences of 510nt to train SpliceBERT on 8 V100 GPUs for a week and the accuracy of MLM task finally reached about 0.56. Then, we continued to pre-train SpliceBERT with sequences ranging from 64nt to 1024nt until the MLM accuracy reached 0.56 again.

### Measuring the impact of genetic variants on their neighbouring nucleotides with KL-divergence

The MLM task requires SpliceBERT to give the probability distribution of each nucleotide's base type based on contextual pre-mRNA sequences. For simplicity, the distribution can be denoted as a 4-dimensional array: $P = (p_A, p_C, p_G, p_T)$ , where the sum of $p_A$, $p_C$, $p_G$ and $p_T$ is 1. Therefore, a genetic variant occurring in a pre-mRNA sequence may change the probability distribution of a nucleotide in this sequences from $P^{ref} = (p_A^{ref}, p_C^{ref}, p_G^{ref}, p_T^{ref})$ to $P^{alt} = (p_A^{alt}, p_C^{alt}, p_G^{alt}, p_T^{alt})$. We defined the impact score $(s)$ to quantify the impact of a variant on a nucleotide with KL-divergence:

$$s = D_{KL}(P^{alt}||P^{ref}) = \sum_{x \in \{A,C,G,T\}} P(x^{alt}) \, log \frac{P(x^{alt})}{P(x^{ref})}.$$

To avoid division by zero error, the probability values were clipped to be between $[10^{-6}, 1]$.

### Estimating splice site usage level from RNA-seq samples

We downloaded the alignment files in bam format of polyA plus RNA-seq and total RNA-seq samples of the K562 cell line. Since there are multiple experiments for each cell line, we retained only the samples of reverse-stranded specificity and discarded samples labelled with "low read depth", "insufficient read depth" or "unreplicated experiment". In total, we obtained 18 K562 samples. See Table S2 for the names and the accession IDs of these files. We removed the reads mapped to multiple locations from the original bam files by keeping only the reads whose mapping quality (MAPQ) equals to 255 (12) using samtools (25). Next, we employed regtools (v0.5.2) (26) to extract the

splice junctions from the bam files and then used SpliSER (v0.1.7) (27) to estimate the splice site usage (SSE) in each sample. The median value of SSEs across multiple samples were taken to annotate the splice sites included in GENCODE (28) annotations (v41).

## Visualization of nucleotide embeddings

For SpliceBERT/DNABERT, we extracted the hidden states of the last Transformer layer as the nucleotide embeddings. For one-hot encoded sequences, we took the nucleotides of our interest together with their flanking sequences within a certain range and used their flattened one-hot encodings as nucleotide embeddings. Before visualization, we conducted principal component analysis (PCA) to reduce the embedding dimension to 128 if the dimension of embedding exceeded 128. Then, we adopted the UMAP algorithm for dimension reduction and visualization. The PCA and UMAP visualization were performed with the Scanpy package (v1.9) (29). The length of token in DNABERT and the size of flanking region in one-hot encoding have a large impact on visualization. We compared different token and sequence lengths (Figure S2) and only showed the best choice (k=3 in DNABERT, flanking size = 20nt in one-hot encoding) in the results.

## Attention weight analysis

In each Transformer layer of SpliceBERT, an attention module transforms the input feature map ($\boldsymbol{h} \in R^{L\times d}$) of a sequence into a key and a query ($K, Q \in R^{L\times d'}$) feature map with two learnable matrices $W_q, W_k \in R^{d\times d'}$, respectively:

$$K = h \times W_k$$

, where $d'$ is defined as the ratio of $d$ to the number of attention head. The dot product of Q and K is the attention matrix:

$$A = softmax\left(\frac{QK^T}{\sqrt{d'}}\right) \in R^{L\times L}.$$

The softmax function normalized the attention matrix by row to make the attention weights in each row sums up to 1. Intuitively, each element $a_{ij}$ in A could be regarded as the importance of the $j$-th token to the $i$-th token (note that $a_{ij}$ is not necessarily equal to $a_{ji}$). In our analysis, we simply averaged $a_{ij}$ and $a_{ji}$ to represent the attention weight between $i$ and $j$. When analyzing the attention of donors and acceptors, we took the average of the weights of 2 nucleotides in intron and 1 nucleotide in exon.

## Finetuning SpliceBERT for downstream tasks

In our study, SpliceBERT can be applied to two types of downstream tasks: sequence-level prediction and token-level prediction. For sequence-level prediction, we took the embedding of the [CLS] token as the sequence-level representation and fed it into a two-layer fully connected neural network for prediction. For token-level prediction, we simply fed the hidden states of the last Transformer layers as token-level representations and fed them to a two-layer neural network to make prediction. During finetuning, we used an initial learning rate of 0.00005 and reduced the learning rate using the "get_polynomial_decay_schedule_with_warmup" scheduler provided in the Huggingface transformer package.

We adopted the 5-fold nested cross validation scheme to finetune and evaluate SpliceBERT on the cross-species splice site prediction task and the human branchpoint prediction task. The samples were first split in to 5 folds. In each epoch, each fold was held out as the test dataset in turn and the other 4 folds were split into a training (3 folds) and a validation dataset (1 fold). The average precision score of the 5 validation datasets were used to select the optimal hyperparameters and the number of training epochs. In this way, all the samples could be used for independent test. Notably, we split the samples into different folds by chromosomes to keep the samples from the same chromosome will also be assigned into the same fold. This can help alleviate the problem of overfitting (30, 31).

## In silico mutagenesis analysis

In silico mutagenesis was employed to investigate the effects of genetic variants on their neighboring nucleotides or splice site/branchpoint predictions. Specifically, we exhaustively enumerated all single nucleotide variants (SNVs) in pre-defined regions of sequences and only one SNV was introduced to a sequence each time.

## Visualization of consensus motifs around splice sites and branchpoints using sequence logo

We visualized the consensus motifs around splice sites and branchpoints using the Logomaker package (32). The logarithm of the ratio between observed and expected nucleotide frequency was used for visualization. The observed frequencies of nucleotide were obtained by taking the average occurrence of nucleotide types across different samples for each position, respectively. The expected frequency of each nucleotide type (background frequency) was obtained by taking its average occurrence among all the samples at all the positions.

# RESULTS

## The SpliceBERT model

SpliceBERT is a language model pre-trained on more than 2 million pre-mRNA sequences from 72 vertebrates with the masked language modelling (MLM) task. In the pre-training stage, we randomly masked about 15% of the nucleotides in the sequences and trained the model to fill out the masked tokens (nucleotides) based on sequence context. In this way, we expected the model to learn high-level representations of nucleotides that can capture the logic of pre-mRNA sequences. Subsequently, we applied SpliceBERT a series of downstream tasks associated the RNA splicing (Figure 1A).

The genes of eukaryotic genes consist of exons and introns and the exons of protein-coding genes can further be partitioned to 5 prime untranslated regions (5'UTR), coding sequences (CDS) and 3 prime untranslated regions (3'UTR). Only the CDS regions encode proteins while they make up only a small fraction of the pre-mRNA sequences. Since the coding regions and non-coding regions may exhibit quite distinct characteristics, we compared the performance of MLM task in different genomic function regions (GFRs) using the MLM accuracy (MLM ACC, the fraction of masked tokens correctly filled out by the model) as a metric. The MLM ACC in introns is about 0.57 in both protein-coding genes and long non-coding RNAs (lncRNAs), which is higher than any other regions by at least 0.05 (Figure 1B). This suggested that it may be easier for MLM in non-coding regions compared to coding regions. We suspected that this was caused by the distinct level of repeats in different GFRs and therefore calculated the fraction of repeated regions according to the RepeatMasker annotation. As shown Figure 1B, the CDS regions have the lowest level of repeats and the introns have the highest level of repeats. This indicated that the fraction of repeated regions have a large impact on the MLM task.

It is well known that evolutionary information provides rich information of the function of biological sequences, which plays an important role in many tasks including protein/RNA structure prediction and genomic variant analysis. The evolutionary information of genome sequences can be acquired by multiple sequence alignment across many species. Given the pre-training of SpliceBERT is also involved with a variety of species, we wondered whether the evolutionary information could be captured by the MLM task. To this end, we first randomly selected 1000 510nt fragments from the human transcriptome and extracted the PhastCons100way conservation score for each nucleotide. The nucleotides with a PhastCons(100way) score of "NaN" were discarded and the distribution of PhastCons(100way) scores is shown in Figure 1C. We labeled the nucleotides with PhastCons score greater than 0.8 as "conservative" and the other nucleotides as "non-conservative". Then, we fitted a linear regression (LR) model using the hidden states in SpliceBERT's last Transformer layer to classify conservative and non-conservative nucleotides. For comparison, we also fitted two models using the embeddings from SpliceBERT-human and one-hot encoded sequences (200nt up-/downstream of the sites), respectively. The LR model using SpliceBERT embeddings achieved the highest average precision (AP) score, which is 0.34, and the other two models achieved the AP scores of 0.27 and 0.10, respectively (Figure 1D). The result demonstrated the language model could implicitly capture evolutionary information through MLM training and pre-training on multiple species outperformed single species.

## Language models enable unsupervised prediction of variant effect on splicing

Several studies have employed pre-trained protein/DNA models for zero-shot mutation effect prediction. These methods usually utilized only the difference between the predicted MLM logits of mutant (MT) allele and wild-type (WT) allele to estimate the mutation effect. However, the impact of genetic variations on the MLM logits of their contextual nucleotides was never explored. Here, we first performed in silico mutagenesis to obtain 10000 single nucleotide variants (SNVs) and applied SpliceBERT on the MT and WT sequences to acquire predicted MLM logits and measured the impact of the SNVs using the KL-divergence between WT logits and MT logits. As shown in Figure 2A, we grouped the SNVs into a conservative group and a non-conservative group according to the PhastCons(100way) score. We found that the SNVs occurring at conservative sites tend to induce a larger impact of predicted MLM logits on their neighboring nucleotides. Similarly, when we performed the same analysis on two datasets of splice-disrupting (SDV) and non-splice-disrupting (non-SDV) variants, VexSeq (33) and MFASS (34), a consistent trend can be observed in Figure 2B and Figure 2C that SDVs tend to have a larger impact on the MLM logits of their contexts.

We next compared the sum of KL-divergence in the 100nt neighboring regions for each SNV (denoted by KL-div(context)) to the logit change of masked mutation position (denoted by Δlogit) adopted by previous studies. The difference of KL-div(context) scores between the SDV and the non-SDV groups is more significant (P-value = $1.6 \times 10^{-10}$/$9.7 \times 10^{-62}$ in VexSeq/MFASS) than that of Δlogit scores (P-value=0.06 /$3.9\times 10^{-48}$ in VexSeq/MFASS) (Figure 3D,3E), indicating that taking the context information into consideration may achieve better performance than just using the mutation site. Moreover, we particularly extracted only the KL-divergence at the splice donor and acceptor sites (KL-div(SS)) to take the distance between mutations and splice donors/acceptors into account. A significant difference could also be observed in the KL-div(SS) scores between the SDV and the non-SDV group (P-value= $2.0 \times 10^{-14}$ / $5.2 \times 10^{-67}$ in VexSeq/MFASS).

Finally, we compared SpliceBERT with SpliceBERT-human, DNABERT and several other unsupervised metrics (phastCons/phyloP conservation score, distance from mutation to splice donors/acceptors) on the VexSeq and the MFASS dataset. We noticed that there are 68.2% and 80.7% of the SNVs in splice donors and acceptors are SDVs in VexSeq and MFASS, which are much higher than other SNVS (4% and 3% in both datasets, respectively), which may dominate the results (Figure S1). We thus excluded the variants in donors and acceptors and plotted the precision-recall curves in Figure 2F and 2G. SpliceBERT consistently outperformed the language models pre-trained with human data only (SpliceBERT-human and DNABERT) and conservation scores, demonstrating the superior performance of language models pre-trained on multiple closely related species.

## SpliceBERT embeddings are informative to characterize splice sites

Since splice sites can be predicted from the RNA sequences, we wondered whether the nucleotide embeddings acquired by unsupervised learning could be utilized to identify splice sites. To this end, we randomly selected 10000 canonical splice sites (5000 GT for donors and 5000 AG for acceptors) and 10000 non-splice sites (5000 GT and 5000 AG) from pre-mRNA sequences annotated by GENCODE v41lift37

annotation and visualized their embeddings (the hidden states in the last layer) with UMAP (Figure 3A). As a comparison, we also visualized the DNABERT (k=3, other results shown in Figure S2) embeddings and one-hot encoded representations of these samples in Figure 3B and 3C. We observed that the embeddings of SpliceBERT and DNABERT can clearly distinguish true splice sites and non-splice sites, while all the samples were mixed in a single cluster in one-hot encoding's UMAP. This suggests that language models can capture the high-level representations associated with the distinct characteristics of splice sites.

Moreover, we visualized the splice sites of high and low usage estimated from RNA-seq samples using the same approach. They were more difficult to be distinguished than splice/non-splice sites as they are all splice sites. This time, only the embeddings from SpliceBERT are informative to separate these samples (Figure 3D) but DNABERT and one-hot failed to do so (Figure 3E and Figure 3F).

Though we performed the above analysis with the hidden states from the last layer, it is also of interest to compare them with the hidden states from other layers. As shown in Figure S3 and S4, the embeddings obtained from layer 2 to layer 6 are all informative for clustering the splice sites and non-splice sites. Interestingly, the best clustering performance was achieved by the 4$^{th}$ layer. However, the mechanisms to explain this observation are beyond our study and require further study.

Taken together, the results suggest that SpliceBERT embeddings captures the distinct characteristics of splice sites and are informative to identify splice sites and their strength. Thus, SpliceBERT may potentially facilitate downstream tasks associated with RNA splicing prediction.

## SpliceBERT captures the association between splice donors and acceptors

We next investigated whether the attention weights in SpliceBERT could capture the association between donor and acceptor sites. To this end, we constructed 6 groups of donor/acceptor/random site pairs and compared the attention weights in these groups. The groups are (1) intronic donor-acceptor pair (D-A(intron)): the donor-acceptor pairs from the same introns, (2) exonic donor-acceptor pair: the donor-acceptor pairs from the same exons, (3) unpaired donor-acceptor combinations: donor-acceptor pairs from different intron/exon, (4) donor-donor pairs, (5) acceptor-acceptor pairs and (6) a control group of randomly selected sites in transcripts. Limited by the maximum sequence that SpliceBERT can process, we retained only the pairs within 800nt. To eliminate the potential bias caused by distance (Figure S5), we guaranteed that the distance distribution of randomly-selected samples is approximately uniformly distributed between 100nt and 800nt, where samples of too short distance (<100nt) were also excluded.

We first showed the distribution of attention weights (averaged across all layers) between the nucleotide pairs in 6 groups (Figure 4A). Donor-acceptor pairs from the same introns tend to have higher attention weights than the other groups. Although the attention weights between other donor-acceptor, donor-donor or acceptor-acceptor pairs are higher than the randomly paired control group, they are still significantly lower than the first group's (P-value $< 1 \times 10^{-6}$ by Mann-Whitney test). This is likely because the donors and acceptors from the same introns are connected in mature mRNAs. As exonic regions tend to have higher conservation scores than intronic regions (Figure 4B), the mature mRNAs may be more conserved across species so that SpliceBERT could identify the association between donors and acceptors separated by intronic regions. This also implied that SpliceBERT is able to detect the long-range dependencies in pre-mRNAs.

Moreover, we also examined the patterns of the attention weights in different Transformer layers. We focused on the attention weights between donors and acceptors from the same introns and analyzed the distribution of donor sites' attention weights around acceptors and vice versa. Specifically, we extracted the attention weights of the donors or acceptors and centered them on the corresponding acceptors or donors, respectively. The median attention weights across samples within 100nt flanking regions of splice sites are shown in Figure 4C. We found that the attention weights in the 3$^{rd}$, the 4$^{th}$ and the 5$^{th}$ layer were enriched around acceptors and donors. For instance, the attention weights in acceptor sites in layers 3-5 are 0.0052, 0.0997 and 0.0046, respectively, which are 2.0, 24.7 and 1.8 times that of the background attention weights (average attention weights within the 201nt window. In contrast, in layers 1,2 and 6 the attention weights in acceptors are only 0.8, 0.5 and 0.7 times the background weights. A similar pattern could be observed in the distribution of acceptors' attention weights around donors (Figure 4C and Table S3). To pursue the reason why the attention weights in layers 3-5 could reflect the association between donors and accepters is beyond this study. However, the result suggested that the hidden states in layers 3-5, especially layer 4, are likely to have a closer relationship to RNA splicing, which is also consistent with our findings in the previous section (Figure S4).

## SpliceBERT enables robust splice site prediction across species

Given that SpliceBERT is able to generate representations related to RNA splicing by unsupervised learning, we next applied it to predict splice sites in different species by supervised learning to see if it could achieve state-of-the-art performance. Specifically, we finetuned SpliceBERT on the training data taken from Scalzitti et al's study (35), which consisted of high-quality splice sites in 114 eukaryotic organisms, and then evaluated it on independent datasets from 5 organisms. We followed the same training/validating/testing scheme described in the Spliceator study to make the results comparable to those reported in the paper. We compared SpliceBERT with SpliceBERT-human, SpliceBERT(w/o PT), DNABERT and 4 existing methods (Spliceator (35), SpliceFinder (36), DSSP (37) and MaxEntScan (1) ) on the test datasets. The test datasets were taken from 5 species: H. sapiens (human), D. rerio(fish), D. melanogaster (fly), C. elegans (worm) and A. thaliana (plant). The F1 scores of these models are shown in Figure 5A and Table S4. SpliceBERT significantly outperformed the other methods with the P-values being less than 0.05 by the Wilcoxon test (Table S4). Notably, SpliceBERT consistently exhibited superior performance to the other methods on 4 non-human datasets, reflecting that the model pre-trained on multiple species can be better generalized to different species. The best-performing model on the human data is SpliceBERT-human. This is likely because the determinants of splice sites vary across species to some extent and thus the model pre-trained on only human data may better capture the logic of human splice site determinants. We also performed in silico mutagenesis analysis to compare the putative variant effect to known consensus sequence of splice sites. For variants affecting donors, we found that the variants leading to large predicted score change enriched

around the donors (1nt in exon and 5nt in intron) (Figure 5B), overlapping with the motifs enriched around the donors (Figure 5C). For variants affecting acceptors, only the 3nt region in the 3'end of introns enriched variants that have a large impact.

It is worth noting that we only focused on evaluating the model's ability to mine the determinants of RNA splicing from short sequences (hundreds of nucleotides). The SpliceAI (9) model was not included for comparison because it requires extremely long sequences of more than 10000 nucleotides.

## SpliceBERT improves branchpoint sites prediction

In addition to splice sites, branchpoint is another important group of RNA splicing regulator as they play a critical role in the identification of acceptors. Here, we applied SpliceBERT for predicting branchpoints collected from Mercer et al's study (38). Since most (~92%) of the BPs locate within the 18-44nt upstream of splice acceptors (Figure 5A), we recruited only the BPs within this region as previous studies did (39, 40).

Before finetuning, we visualized the SpliceBERT, DNABERT and one-hot encoding embeddings of BPs and randomly selected non-BPs using the UMAP algorithm for A, C, G and T, respectively. Unlike splice sites, we found that BPs and non-BPs could not be clearly separated in the UMAP of SpliceBERT embeddings (Figure 5B) and the results were even worse for DNABERT and one-hot encoding embeddings (Figure S5). This is likely because human branchpoints are usually degenerative (13) and cannot be well characterized like splice sites.

To predict BPs, we finetuned SpliceBERT to predict the high-confidence BPs in the 18-44nt upstream of splice acceptors. The low-confidence BPs were excluded from the dataset and other sites within the regions were used as negative samples. The 5-fold nested CV strategy was used to train and evaluate the model. We compared SpliceBERT against 5 other models, including SpliceBERT-human, SpliceBERT(w/o PT), DNABERT, LaBranchoR (41) and Branchpointer (39). For a fair comparison, these models (except for Branchpointer) were trained and evaluated under the same scheme as SpliceBERT. The predictions from Branchpointer were directly generated using its scripts. The performance was evaluated using AP scores, and the PR-curves were plotted in Figure 5C. SpliceBERT outperformed all the baseline models, and the AP improvement was between 0.013 and 0.084. Notably, the AP score of SpliceBERT(w/o PT) was only 0.661. Although it has the same architecture as SpliceBERT/SpliceBERT-human, it was trained from scratch for BP prediction without pre-training. The gap between the performance of SpliceBERT(w/o PT) and SpliceBERT/SpliceBERT-human demonstrated the importance of pre-training.

Next, we performed in silico mutagenesis on the regions around true BPs to visualize the effect of genetic variants on BP recognition. The variants between the BP – 3 (3nt upstream of BP) and BP + 1 (1nt downstream of BP) of BPs have the largest putative impact on BP prediction (Figure 5D), exactly matching the sequence consensus of known BPs (Figure 5E). Moreover, we also visualized the change of predicted BP scores induced by specific variant types in Figure 5F. For example, the loss of adenine at the BP sites will significantly decrease the predicted BP scores, and variant T>A and T>G at BP-2 can also largely decrease the BP scores. These observations are consistent with the known pattern of pathogenic variants around BP sites as reported in Zhang et al's study (42).

## DISCUSSION

Genetic information, encoded in the genomes, provided a unique challenge for language models. We are only at the very beginning of unveiling the great learning potential of language models in solving computational genomic problems. Unlike human languages, genomic sequences are extremely long and the nucleotide composition in different functional loci may have completely different characteristics (e.g., different types of repetitive regions), which may bring critical challenges to directly apply the models for human languages successfully. Therefore, in this study, we focused pre-mRNAs and developed SpliceBERT pre-trained on vertebrate pre-mRNA sequences to facilitate the studies of RNA splicing. We were particular interest in how to connect pre-trained models and biological knowledge to give explainable evidence why pre-trained LLMs could improve sequence-based RNA prediction.

Multiple sequence alignment of genome/protein sequences across multiple species has shown a substantial capacity to improve sequence-based predictions because it could capture evolutionary information hidden in the sequences. In our study, we found that SpliceBERT pre-trained on multiple species exhibited superior power to single-species DNA/RNA language models in preserving the evolutionary information in nucleotide embeddings. It suggested that although SpliceBERT did not explicitly perform MSA during pre-training, the MLM task may also play a role similar to MSA to obtain evolutionary information. In addition, we found that the nucleotide embeddings generated by SpliceBERT, pre-trained by unsupervised learning only, not only provide information for identifying splice sites, but also have the potential to distinguish strong and weak splice sites. More importantly, the donor-acceptor pairs from the same introns tend to have higher attention weights than other types of donor and acceptor pairs. This indicated that nucleotides' distinct properties associated with RNA splicing can be captured by unsupervised learning on the primary sequences of transcripts. Interestingly, we also noticed that splice-disrupting variants tend to induce a larger impact on the embeddings of their flanking nucleotides than benign ones, which was also useful to identify pathogenic mutations. When we finetuned SpliceBERT to predict splice sites in multiple species and human branchpoints, it outperformed conventional baseline models as well as language models pre-trained only on human data. Our analysis and experiments indicated the effectiveness of multi-species unsupervised learning for sequence-based modelling of RNA splicing.

Currently, we only applied SpliceBERT to predict splice sites and branchpoints independent of tissue or cell types, while tissue/cell type-specific alternative splicing cannot be ignored because they contributed a lot to transcriptomic and proteomic diversity (43). In fact, predicting tissue/cell type-specific splicing is still challenging solely from sequences (10, 44). We may combine additional tasks like tissue-specific splice site strength (10) and RNA binding protein binding sites (45) to improve the model's ability for tissue-specific prediction.

Another potential limitation of SpliceBERT is that it could only process sequences no longer than 1024nt because the memory complexity of vanilla Transformer layers is quadratic to the sequence length. However, in many tasks in genomics, including splice site prediction, integrating large genomic context from thousands to millions of nucleotides is crucial to ensure the accuracy of predictions (9, 46–48). One straightforward solution is to adopt Transformers with linear memory complexity with respect to sequence length. Besides, we may employ distill learning techniques to transfer the knowledge in LLMs to lightweight architectures like CNN to handle long sequences. Besides, given recent studies have shown that some improved versions of CNNs can achieve superior performance to Transformer-based models (49, 50), it is also promising to develop pre-trained genomic sequence language models without Transformer (51).

## Conclusion

We developed SpliceBERT, a pre-trained language model for pre-mRNA sequences to study and improve sequence-based prediction of RNA splicing. SpliceBERT successfully captured evolutionary information and RNA splicing signals from primary sequences by unsupervised learning. Downstream tasks further demonstrated that our model outperformed the other language models pre-trained on only human data. Our study provided a better understanding of the possible mechanisms by which pre-trained models could improve sequence-based predictions of RNA splicing.

## AVAILABILITY

The source codes and pre-trained model weights of SpliceBERT is available at https://github.com/biomed-AI/SpliceBERT .

## FUNDING

The research was supported by National Key R&D Program of China (2020YFB0204803); National Natural Science Foundation of China (12126610); Guangdong Key Field R&D Plan (2019B020228001 and 2018B010109006); Introducing Innovative and Entrepreneurial Teams (2016ZT06D211); Guangzhou S&T Research Plan (202007030010); the Peng Cheng Laboratory grant PCL2021A13 and by Peng Cheng Cloud-Brain.

### CONFLICT OF INTEREST

None declared

## REFERENCES

1. Yeo,G. and Burge,C.B. (2004) Maximum entropy modeling of short sequence motifs with applications to RNA splicing signals. *J. Comput. Biol.*, **11**, 377–394.

2. Barash,Y., Calarco,J.A., Gao,W., Pan,Q., Wang,X., Shai,O., Blencowe,B.J. and Frey,B.J. (2010) Deciphering the splicing code. *Nature*, **465**, 53.

3. Xiong,H.Y., Alipanahi,B., Lee,L.J., Bretschneider,H., Merico,D., Yuen,R.K.C., Hua,Y., Gueroussov,S., Najafabadi,H.S., Hughes,T.R., *et al.* (2015) The human splicing code reveals new insights into the genetic determinants of disease. *Science*, **347**, 1254806–1254806.

4. Cartegni,L., Wang,J., Zhu,Z., Zhang,M.Q. and Krainer,A.R. (2003) ESEfinder: A web resource to identify exonic splicing enhancers. *Nucleic Acids Res*, **31**, 3568–3571.

5. Zhang,Q., Fan,X., Wang,Y., Sun,M., Shao,J. and Guo,D. (2017) BPP: a sequence-based algorithm for branch point prediction. *Bioinformatics*, **33**, 3166–3172.

6. Desmet,F.-O., Hamroun,D., Lalande,M., Collod-Béroud,G., Claustres,M. and Béroud,C. (2009) Human Splicing Finder: an online bioinformatics tool to predict splicing signals. *Nucleic Acids Research*, **37**, e67–e67.

7. Rosenberg,A.B., Patwardhan,R.P., Shendure,J. and Seelig,G. (2015) Learning the Sequence Determinants of Alternative Splicing from Millions of Random Sequences. *Cell*, **163**, 698–711.

8. Cheng,J., Nguyen,T.Y.D., Cygan,K.J., Çelik,M.H., Fairbrother,W.G., Avsec, žiga and Gagneur,J. (2019) MMSplice: modular modeling improves the predictions of genetic variant effects on splicing. *Genome Biology*, **20**, 48.

9. Jaganathan,K., Kyriazopoulou Panagiotopoulou,S., McRae,J.F., Darbandi,S.F., Knowles,D., Li,Y.I., Kosmicki,J.A., Arbelaez,J., Cui,W., Schwartz,G.B., *et al.* (2019) Predicting Splicing from Primary Sequence with Deep Learning. *Cell*, **176**, 535-548.e24.

10. Zeng,T. and Li,Y.I. (2022) Predicting RNA splicing from DNA sequence using Pangolin. *Genome Biology*, **23**, 103.

11. Trapnell,C., Williams,B.A., Pertea,G., Mortazavi,A., Kwan,G., van Baren,M.J., Salzberg,S.L., Wold,B.J. and Pachter,L. (2010) Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nat Biotechnol*, **28**, 511–515.

12. Dobin,A., Davis,C.A., Zaleski,C., Schlesinger,F., Drenkow,J., Chaisson,M., Batut,P., Jha,S. and Gingeras,T.R. (2012) STAR: ultrafast universal RNA-seq aligner. *Bioinformatics*, **29**, 15–21.

13. Gao,K., Masuda,A., Matsuura,T. and Ohno,K. (2008) Human branch point consensus sequence is yUnAy. *Nucleic Acids Research*, **36**, 2257–2267.

14. Devlin,J., Chang,M.-W., Lee,K. and Toutanova,K. (2019) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, pp. 4171–4186.

15. Rives,A., Meier,J., Sercu,T., Goyal,S., Lin,Z., Liu,J., Guo,D., Ott,M., Zitnick,C.L., Ma,J., *et al.* (2021) Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, **118**, e2016239118.

16. Elnaggar,A., Heinzinger,M., Dallago,C., Rehawi,G., Wang,Y., Jones,L., Gibbs,T., Feher,T., Angerer,C., Steinegger,M., *et al.* (2021) ProtTrans: Towards Cracking the Language of Lifes Code Through Self-Supervised Deep Learning and High Performance Computing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10.1109/TPAMI.2021.3095381.

17. Chen,J., Hu,Z., Sun,S., Tan,Q., Wang,Y., Yu,Q., Zong,L., Hong,L., Xiao,J., King,I., *et al.* (2022) Interpretable RNA Foundation Model from Unannotated Data for Highly Accurate RNA Structure and Function Predictions. 10.48550/arXiv.2204.00300.

18. Zvyagin,M., Brace,A., Hippe,K., Deng,Y., Zhang,B., Orozco Bohorquez,C., Clyde,A., Kale,B., Perez-Rivera,D., Ma,H., *et al*. (2022) GenSLMs: Genome-scale language models reveal SARS-CoV-2 evolutionary dynamics Bioinformatics.

19. Ji,Y., Zhou,Z., Liu,H. and Davuluri,R.V. (2021) DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome. *Bioinformatics*, **37**, 2112–2120.

20. Yang,M., Huang,L., Huang,H., Tang,H., Zhang,N., Yang,H., Wu,J. and Mu,F. (2022) Integrating convolution and self-attention improves language model of human genome for interpreting non-coding regions at base-resolution. *Nucleic Acids Research*, 10.1093/nar/gkac326.

21. Haeussler,M., Zweig,A.S., Tyner,C., Speir,M.L., Rosenbloom,K.R., Raney,B.J., Lee,C.M., Lee,B.T., Hinrichs,A.S., Gonzalez,J.N., *et al.* (2019) The UCSC Genome Browser database: 2019 update. *Nucleic Acids Res.*, **47**, D853–D858.

22. Quinlan,A.R. and Hall,I.M. (2010) BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*, **26**, 841–842.

23. Ng,P. (2017) dna2vec: Consistent vector representations of variable-length k-mers. *arXiv:1701.06279 [cs, q-bio, stat]*.

24. Cahyawijaya,S., Yu,T., Liu,Z., Zhou,X., Mak,T.W.T., Ip,Y.Y.N. and Fung,P. (2022) SNP2Vec: Scalable Self-Supervised Pre-Training for Genome-Wide Association Study. In *Proceedings of the 21st Workshop on Biomedical Language Processing*. Association for Computational Linguistics, Dublin, Ireland, pp. 140–154.

25. Li,H., Handsaker,B., Wysoker,A., Fennell,T., Ruan,J., Homer,N., Marth,G., Abecasis,G. and Durbin,R. (2009) The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, **25**, 2078–2079.

26. Cotto,K.C., Feng,Y.-Y., Ramu,A., Skidmore,Z.L., Kunisaki,J., Richters,M., Freshour,S., Lin,Y., Chapman,W.C., Uppaluri,R., *et al.* (2021) RegTools: Integrated analysis of genomic and transcriptomic data for the discovery of splicing variants in cancer. 10.1101/436634.

27. Dent,C.I., Singh,S., Mukherjee,S., Mishra,S., Sarwade,R.D., Shamaya,N., Loo,K.P., Harrison,P., Sureshkumar,S., Powell,D., *et al.* (2021) Quantifying splice-site usage: a simple yet powerful approach to analyze splicing. *NAR Genomics and Bioinformatics*, **3**, lqab041.

28. Frankish,A., Diekhans,M., Ferreira,A.-M., Johnson,R., Jungreis,I., Loveland,J., Mudge,J.M., Sisu,C., Wright,J., Armstrong,J., *et al.* (2019) GENCODE reference annotation for the human and mouse genomes. *Nucleic Acids Res.*, **47**, D766–D773.

29. Wolf,F.A., Angerer,P. and Theis,F.J. (2018) SCANPY: large-scale single-cell gene expression data analysis. *Genome Biology*, **19**, 15.

30. Whalen,S., Schreiber,J., Noble,W.S. and Pollard,K.S. (2022) Navigating the pitfalls of applying machine learning in genomics. *Nat Rev Genet*, **23**, 169–181.

31. Cao,F. and Fullwood,M.J. (2019) Inflated performance measures in enhancer–promoter interaction-prediction methods. *Nat Genet*, **51**, 1196–1198.

32. Tareen,A. and Kinney,J.B. (2020) Logomaker: beautiful sequence logos in Python. *Bioinformatics*, **36**, 2272–2274.

33. Adamson,S.I., Zhan,L. and Graveley,B.R. (2018) Vex-seq: high-throughput identification of the impact of genetic variation on pre-mRNA splicing efficiency. *Genome Biology*, **19**, 71.

34. Cheung,R., Insigne,K.D., Yao,D., Burghard,C.P., Wang,J., Hsiao,Y.-H.E., Jones,E.M., Goodman,D.B., Xiao,X. and Kosuri,S. (2019) A Multiplexed Assay for Exon Recognition Reveals that an

Unappreciated Fraction of Rare Genetic Variants Cause Large-Effect Splicing Disruptions. *Molecular Cell*, **73**, 183-194.e8.

35. Scalzitti,N., Kress,A., Orhand,R., Weber,T., Moulinier,L., Jeannin-Girardon,A., Collet,P., Poch,O. and Thompson,J.D. (2021) Spliceator: multi-species splice site prediction using convolutional neural networks. *BMC Bioinformatics*, **22**, 561.

36. Wang,R., Wang,Z., Wang,J. and Li,S. (2019) SpliceFinder: ab initio prediction of splice sites using convolutional neural network. *BMC Bioinformatics*, **20**, 652.

37. Naito,T. (2018) Human Splice-Site Prediction with Deep Neural Networks. *J Comput Biol*, **25**, 954–961.

38. Mercer,T.R., Clark,M.B., Andersen,S.B., Brunck,M.E., Haerty,W., Crawford,J., Taft,R.J., Nielsen,L.K., Dinger,M.E. and Mattick,J.S. (2015) Genome-wide discovery of human splicing branchpoints. *Genome Res*, **25**, 290–303.

39. Signal,B., Gloss,B.S., Dinger,M.E. and Mercer,T.R. (2018) Machine learning annotation of human branchpoints. *Bioinformatics*, **34**, 920–927.

40. Avsec,Ž., Barekatain,M., Cheng,J. and Gagneur,J. (2018) Modeling positional effects of regulatory sequences with spline transformations increases prediction accuracy of deep neural networks. *Bioinformatics*, **34**, 1261–1269.

41. Paggi,J.M. and Bejerano,G. (2018) A sequence-based, deep learning model accurately predicts RNA splicing branchpoints. *RNA*, **24**, 1647–1658.

42. Zhang,P., Philippot,Q., Ren,W., Lei,W.-T., Li,J., Stenson,P.D., Palacín,P.S., Colobran,R., Boisson,B., Zhang,S.-Y., *et al.* (2022) Genome-wide detection of human variants that disrupt intronic branchpoints. *Proceedings of the National Academy of Sciences*, **119**, e2211194119.

43. Ling,J.P., Wilks,C., Charles,R., Leavey,P.J., Ghosh,D., Jiang,L., Santiago,C.P., Pang,B., Venkataraman,A., Clark,B.S., *et al.* (2020) ASCOT identifies key regulators of neuronal subtype-specific splicing. *Nat Commun*, **11**, 137.

44. Cheng,J., Çelik,M.H., Kundaje,A. and Gagneur,J. (2021) MTSplice predicts effects of genetic variants on tissue-specific splicing. *Genome Biology*, **22**, 94.

45. Horlacher,M., Wagner,N., Moyon,L., Kuret,K., Goedert,N., Salvatore,M., Ule,J., Gagneur,J., Winther,O. and Marsico,A. (2022) Towards In-Silico CLIP-seq: Predicting Protein-RNA Interaction via Sequence-to-Signal Learning. 10.1101/2022.09.16.508290.

46. Schwessinger,R., Gosden,M., Downes,D., Brown,R.C., Oudelaar,A.M., Telenius,J., Teh,Y.W., Lunter,G. and Hughes,J.R. (2020) DeepC: predicting 3D genome folding using megabase-scale transfer learning. *Nat Methods*, **17**, 1118–1124.

47. Chen,K., Zhao,H. and Yang,Y. (2022) Capturing large genomic contexts for accurately predicting enhancer-promoter interactions. *Briefings in Bioinformatics*, **23**, bbab577.

48. Lee,D., Yang,J. and Kim,S. (2022) Learning the histone codes with large genomic windows and three-dimensional chromatin interactions using transformer. *Nat Commun*, **13**, 6678.

49. Liu,Z., Mao,H., Wu,C.-Y., Feichtenhofer,C., Darrell,T. and Xie,S. (2022) A ConvNet for the 2020s. 10.48550/arXiv.2201.03545.

50. Woo,S., Debnath,S., Hu,R., Chen,X., Liu,Z., Kweon,I.S. and Xie,S. (2023) ConvNeXt V2: Co-designing and Scaling ConvNets with Masked Autoencoders.

51. Benegas,G., Batra,S.S. and Song,Y.S. (2022) DNA language models are powerful zero-shot predictors of non-coding variant effects. 10.1101/2022.08.22.504706.

## TABLE AND FIGURES LEGENDS

**Figure 1**

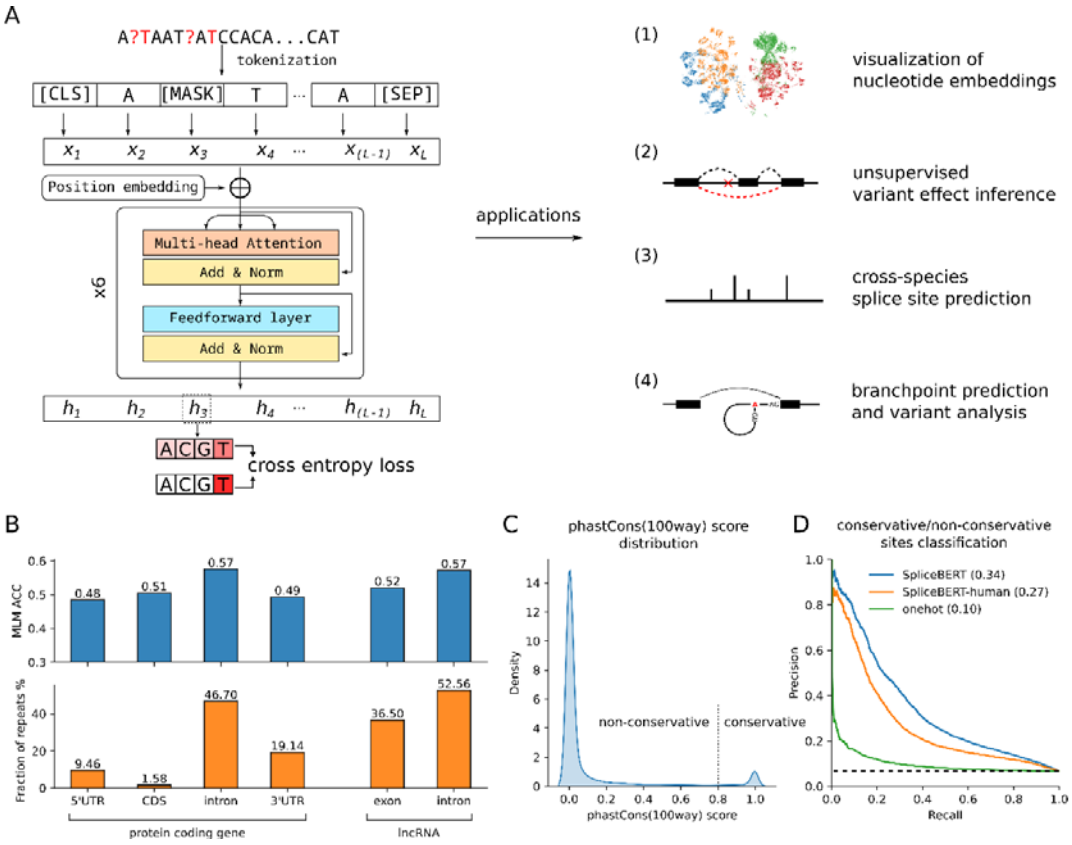


**Figure 1.** An overview of SpliceBERT. (A) SpliceBERT was pre-trained on the pre-mRNA sequences of 72 vertebrates under the masked language modeling (MLM) framework. We additionally applied SpliceBERT for downstream tasks of (a) unsupervised variant effect prediction, (b) visualization of nucleotide embeddings, (c) cross-species splice sites prediction and (d) branchpoints prediction and variant analysis. (B) The accuracy of masked language modelling (MLM ACC) of SpliceBERT and the fraction of repeats in different genomic function regions. (C) The distribution of phastCons(100way) score of randomly selected sites in pre-mRNA sequences. (D) The precision-recall curves of linear regression (LR) models for predicting conservative/non-conservative sites based on SpliceBERT, SpliceBERT-human and one-hot encoded nucleotide embeddings.

**Figure 2**

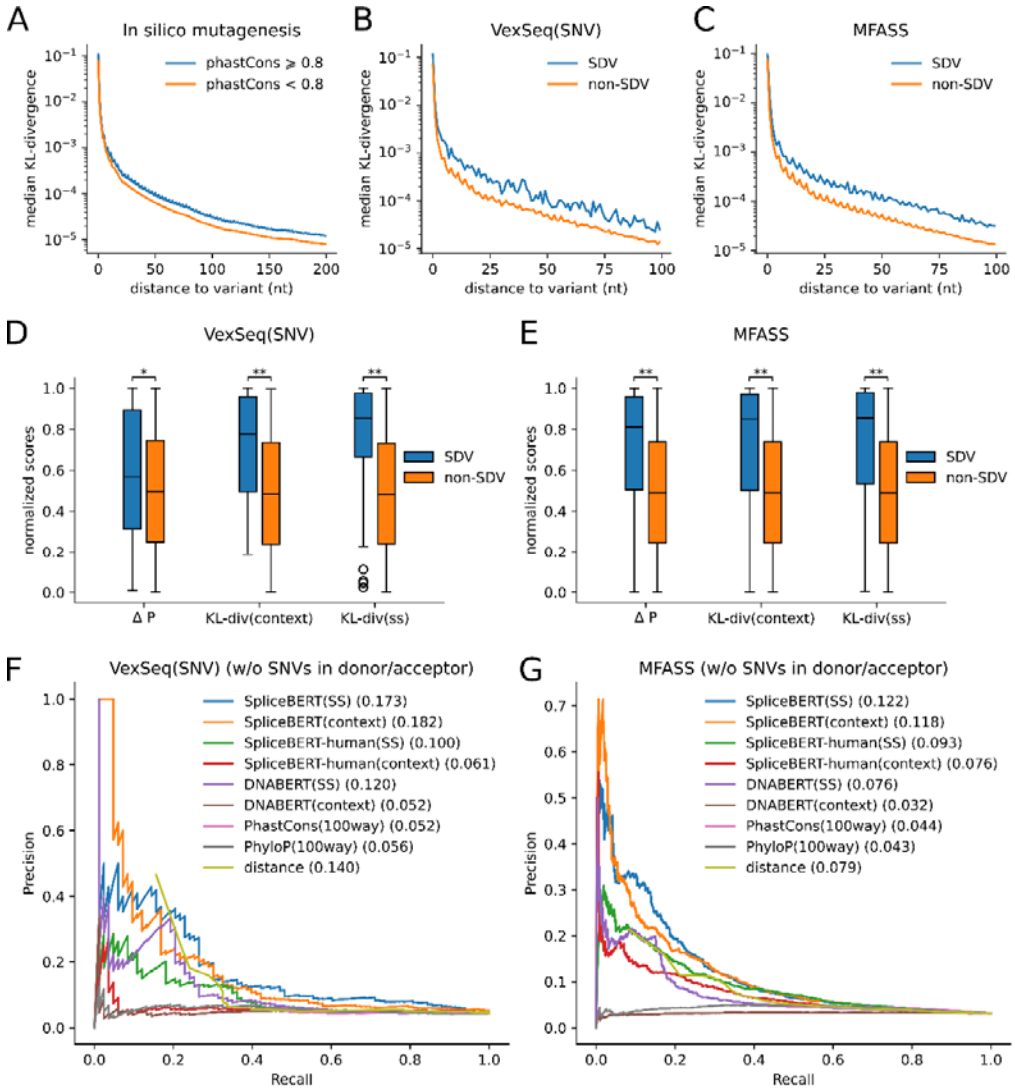


**Figure 2.** SpliceBERT enables unsupervised splice-disrupting variant prediction. The relationship between the median value of KL-divergence between the MLM logits of wild-type sequences and mutant sequences and the distances to variants in (A) conservative and non-conservative sites sampled by in silico mutagenesis, (B) VexSeq SDV (splice-disrupting variant) and non-SDV (non-splice-disrupting variant) group, and (C) MFASS SDV and non-SDV group. The boxplot of delta-logit ( logit), the sum of KL-divergence in neighboring nucleotides (KL-div(context)) and the KL-divergence at splice donors/acceptors (KL-div(SS)) of the SDV and non-SDV group in (D) VexSeq and (E) MFASS. The precision-recall curves of SpliceBERT and baseline methods for classifying SDV and non-SDV samples in (F) VexSeq and (G) MFASS.

**Figure 3**

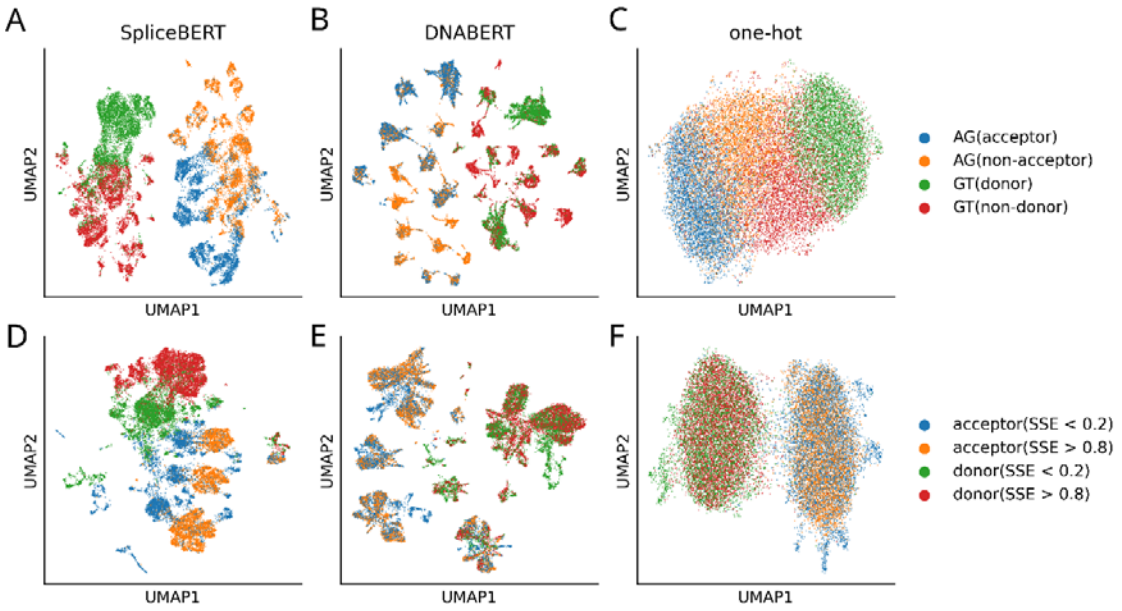


**Figure 3.** UMAP visualization of splice site embeddings. Visualization of canonical splice sites and non-splice sites with GT and AG dinucleotides based on (A) SpliceBERT embeddings, (B) DNABERT(k=3) embeddings and (C) one-hot encodings. Visualization of splice sites of high and low usage in K562 cell line estimated by SpliSER based on (D) SpliceBERT embeddings, (E) DNABERT(k=3) embeddings or (F) one-hot encodings.

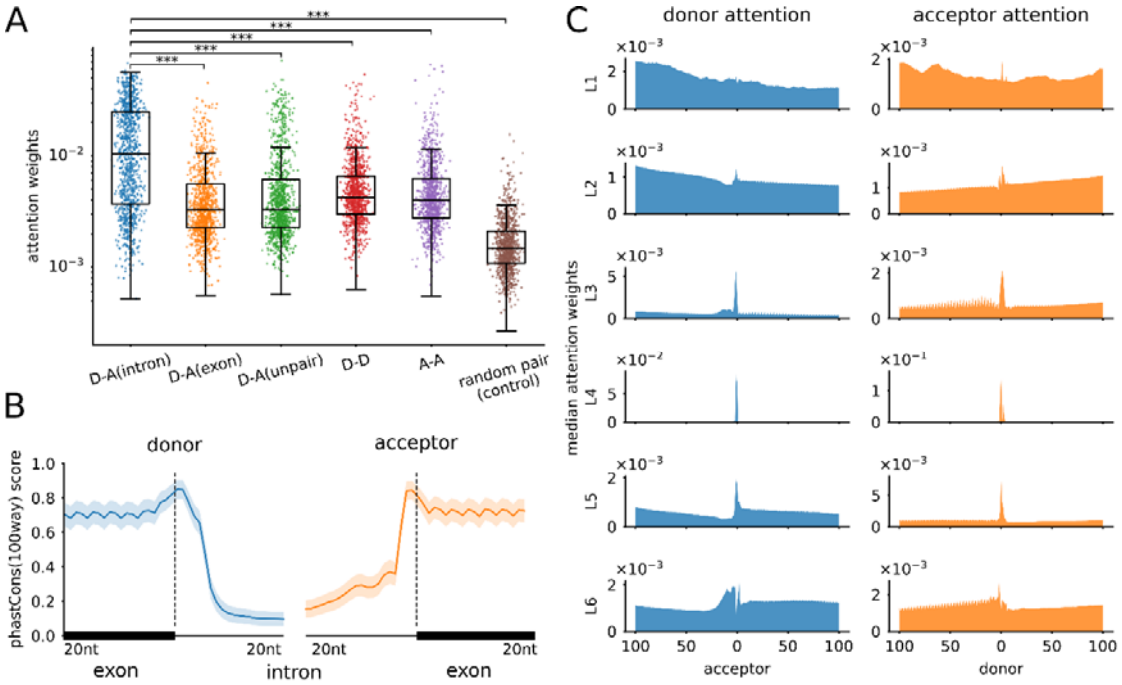


**Figure 4**

**Figure 4.** Analysis of the attention weights in SpliceBERT. **(A)** The distribution of attention weights between donors and acceptors from the same intron (D-A(intron)), the same exon (D-A(exon)), different intron/exon (D-A(unpair)), between donors and donors (D-D) and between acceptors and acceptors (A-A). A control group of randomly paired nuelcotides is also shown (random pair). The average attention weights across all the 6 layers in SpliceBERT were used. The statistical significance was assessed with t-test. **(B)** The distribution of phastCons(100way) scores around donors and acceptors. The average value and standard error of the conservations scores in 1000 randomly selected donor-acceptor pairs (from the same introns) are shown. **(C)** The distribution of donors' and acceptor's atteiton around acceptors and donors, respectively, in each Transformer layer. The median attention weights across different samples were used. (D-A: donor-acceptor, D-D: donor-donor, A-A: acceptor-acceptor, L: layer. ***: P-value <  )

**Figure 5**

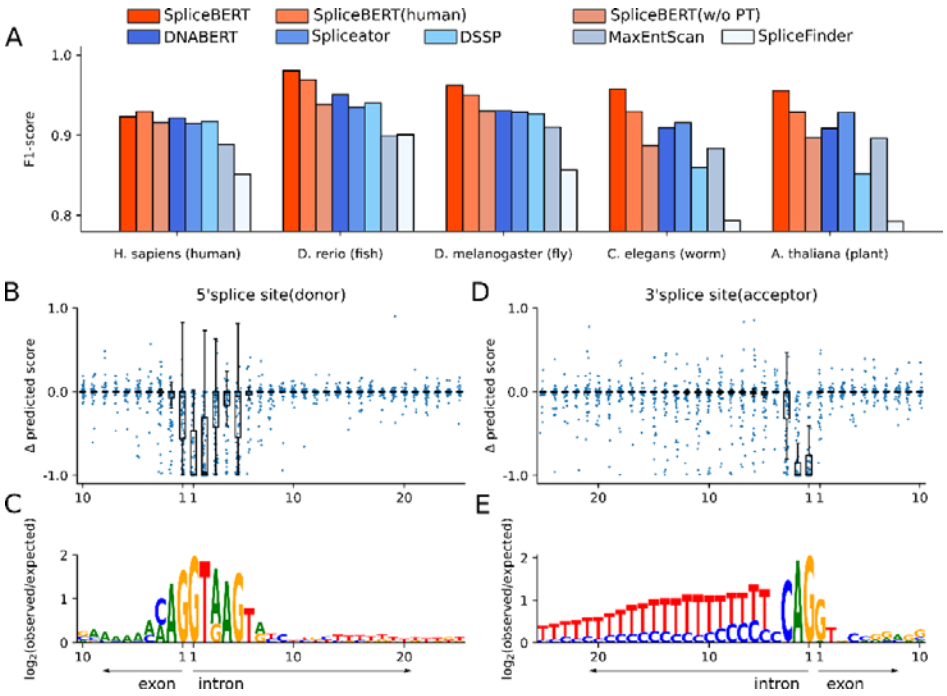


**Figure 5.** SpliceBERT predicts splice sites in different species. (A) The F1-score of SpliceBERT and baseline models on 5 different species. The visualization of the impact of in silico mutagenesis variants on splice site scores for (B) donors and (D) acceptors and corresponding sequence logos of wild-type sequences (C and E).

## Figure 6

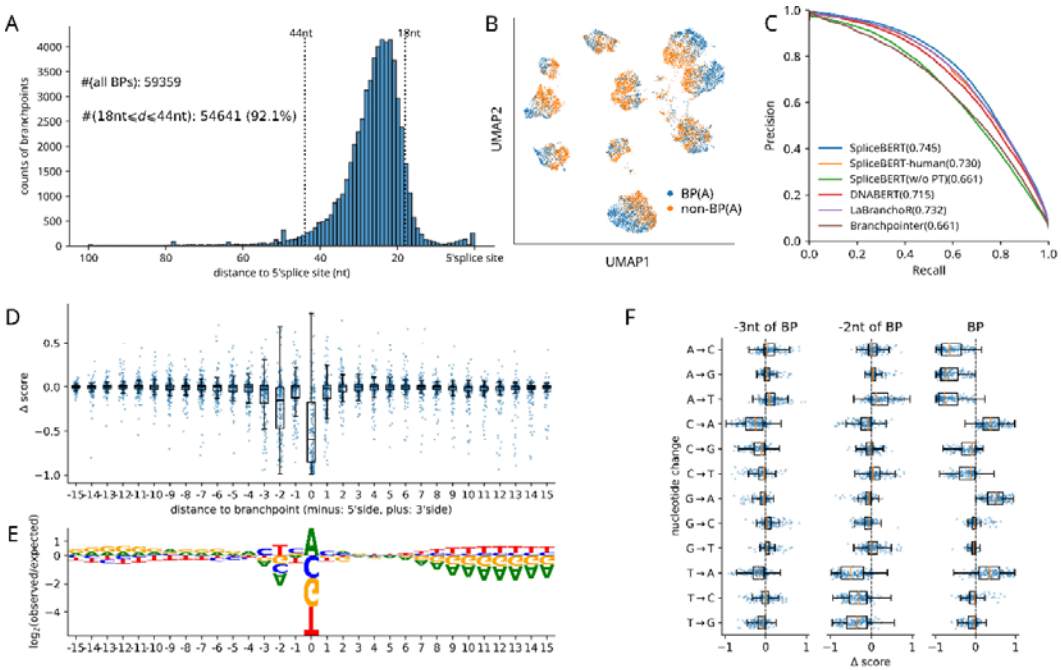


**Figure 6.** SpliceBERT predicts human branchpoint sites. (A) The distribution of high-confidence branchpoints in the Mercer et al dataset. (B) The UMAP visualization of branchpoint and non-branchpoint site embeddings (only adenine sites are shown). (C) The precision-recall curves of the models for predicting branchpoints. (D) The impact of variants at different sites on SpliceBERT predicted BP scores and (E) the corresponding logos of wild-type sequences. (F) The predicted BP score change induced by different types of mutations at BP sites and their 2nt- and 3nt- upstream.